Title: Trend Toward Flexible Management Practice in the Computer Industry

Course: EMGT 520/620
Term: Fall
Year: 1998
Author(s): S. Barnum, G. Austin, Y. Daim, Z. Cursetjee, S. Premjai, A. Khanna, M. Thavasi and R. G.

Report No: P98069

Abstract: This paper briefly describes the older, more rigid practices and contrasts this to the goals of the new solutions. The scope is contained to what many companies are doing as far as marketing strategy, operational strategy, and engineering life-cycle in an attempt to not only remain competitive but also to be market leaders.

Trend Toward Flexible Management Practice in the Computer Industry

S. Barnum, G. Austin, Y. Daim, Z. Cursetjee, S. Premjai, A. Khanna, M. Thavasi, G. Rousseau.

EMP-P98069

Table of Contents

Environment

The computer industry is changing rapidly. Companies face fierce competition as they strive to decrease time to market, yet remain flexible to meet customer demands; lower product cost and ensure high quality products. To manage this environment, several companies have changed the way they market and develop their products.

In effect, the computer industry is moving away from more centralized, compartmentalized, and rigid practices and structures toward more distributed, integrated, and flexible practices and structures. Examples of this trend can be observed within a company's marketing strategy (for example, focusing on the consumer with direct sales of build-to-order PCs), operational strategy (for example, applying the concept of reuse), and engineering life-cycle (for example, performing concurrent engineering and iterative and incremental product development).

This paper briefly describes the older, more rigid practices and contrasts this to the goals of the new solutions. The scope is contained to what many companies are doing as far as marketing strategy, operational strategy, and engineering life-cycle in an attempt to not only remain competitive but also to be market leaders. Examples are provided as to how the new practices are being implemented by many companies such as

- Compaq
- Dell
- Ericsson
- Gateway
- Microsoft
- Motorola
- Netscape
- Synopsys

within various sectors of the computer industry including

- chips
- computers
- operating systems
- application software
- sales/distribution.

Specific trends discussed are:

Area of Focus	Trend
Marketing Strategy	A move away from engineering-driven product development (i.e., the company builds a product based on what the company believes the market will want) toward customer-driven product development (i.e., the company seeks to understand what its customers need/want, then the company builds a product to meet those needs).
Operational Strategy	A move away from independent, custom development (i.e., each project and product is built from the ground up) toward modular, reusable development (i.e., projects and products are planned and designed in such a way that effort and resources expended on one project or product can be reused on other projects and products).
Engineering Life-Cycle	A move away from rigid, compartmentalized waterfall life-cycles (i.e., marketing develops the concept and requirements, then an engineering architecture team performs the entire design, then development engineers implement the entire design, then a testing team tests the entire design, then a manufacturing or deployment team builds the product and delivers it to the customer) toward more flexible, efficient approaches such as concurrent engineering and iterative and incremental life-cycles (i.e., the above waterfall process is actually performed start to finish several times throughout a project where each pass through looks at a smaller chunk of the overall functionality).

Key Issues / Goals

Marketing Strategy

Until the 1980s, many companies tended to use product-driven marketing. This involved developing a product, through research and development, that was considered to be the furthest advance within the field. The product was typically developed independent of specific market needs. The mindset was, "We will develop the best and everyone will want to buy it." IBM used the product-driven approach successfully for more than 50 years, until smaller, more agile competitors moved to customer-driven marketing and IBM could no longer keep pace. [5]

Customer-driven marketing is one of the many buzzwords in corporate marketing. Simply put, this strategy means being customer-driven in terms of product development, product marketing position, product pricing, and customer support. Most successful companies try first to determine exactly what the customer wants before delivering a product. Being customer-driven applies both to companies that supply tangible goods and those that supply intangibles. [32]

"All other things being equal, customer-driven marketing will always conquer." [5] Specific examples of customer-driven marketing success stories are discussed within the Marketing Strategy section.

Operational Strategy

A company's operational strategy determines the direction the team members should take when planning projects and/or creating products in response to business needs. Especially in the area of applications development, there has been and continues to be a tendency within some companies to define projects and build products from the ground up. In other words, each project or product is viewed as independent of another and, as a result, people tend to "reinvent the wheel" each time.

There is a fundamental shift occurring in the operational strategy of companies in the computer industry. This shift is characterized by a movement away from the past --- independent and fully customized product development --- toward a strategy of utilizing modular and reusable components. This is most clearly exhibited in the form of object-oriented efforts in software development and is also making its way into other sectors of the computer industry.

The benefits resulting from strategic reuse include decreased time-to-market, higher quality products, and lower overall development costs. Specific examples of strategic reuse success stories are discussed within the Operational Strategy section.

Engineering Life-Cycle

The dynamic environment of the computer industry has provided serious challenges to today's standard engineering management practices. Driven by the growth of global networks and other integrated information technologies we see an increasing compression of development timelines. Gone are the days when companies competed for the market based primarily on functionality and quality. Today the first product to market has an overwhelming advantage, and to fight for this honor requires that companies seriously rethink the way they develop their products. Further, these compressed timelines not only require shorter overall development cycles but also affect the very nature of product development. In the past the customer was usually involved only at both ends of the development cycle, for requirements development and product acquisition. Today, with communication timelines compressed by advanced information technologies and information instantly accessible to the customer, we see the customer demanding a voice throughout product development including feedback on product status. In this highly charged competitive atmosphere flexibility has become the key to success. The Engineering Life-Cyle section approaches the changing environment from an engineering process perspective, briefly presenting the current standard practices and then discussing the trend for future solutions.

Marketing Strategy

An analysis of the PC industry is found within this section. More specifically, this section focuses on the major players and how they are planning their marketing strategy in an attempt to dominate or survive in a very dynamic, competitive marketplace. The two main aspects discussed in this section are: (1) the changing distribution model and (2) what the post-PC era might bring.

The Major Players in the PC Industry

Major players within the PC industry include Dell, Compaq, Gateway, HP and IBM. Dell and Compaq are leading the pack of PC manufacturers for market shares, Dell with 12.7% and Compaq with 12.3% in the second quarter 1998. Gateway is ranked third at 7.2%. [23]

In the consumer market, Dell only had 2.4% market share last year [27] with Compaq, Gateway and Packard-Bell holding 50%. Dell receives 80% of its overall revenue from selling to businesses and institutions, with "retail" sales to individuals making up the rest. HP has a 32.5% retail market share and leads this segment with Compaq, IBM and Packard-Bell. Apple lags fifth with a 4.4% market share [1]. Apple maintains a strong position in the education and desktop publishing markets; however, its presence in the higher-growth corporate and consumer markets is weak and declining.

The Supply Chain as a Strategic Source of Value

"The way customers purchase computers is changing," admits Compaq Computer Corp. "Some customers want to buy products that will remain consistent in order to minimize the life-cycle costs associated with supporting and maintaining the installed base of PCs. Other customers want to take advantage of changes in technology to make the best purchasing decisions at any given time." [37]

The major PC manufacturers are moving toward 'build-to-order' rather than 'build-to-forecast.' Teresko writes, "Not too long ago, PC vendors followed a simpler, more traditional marketing model: first forecast demand and then push finished goods into markets via multi-tiered channels with many players. Although that model continually suffered from inaccurate forecasts, business survival was possible just by churning out the hottest new technologies every 18 months or so." The traditional model worked quite well until Michael Dell came along in 1984 and introduced a new way to compete --- not by technology alone, but by emphasizing the needs of the customers with an ability to satisfy and serve them quickly and efficiently. Dell put pressure on the industry's traditional players with a simple

concept: sell personal computers directly to customers with no complicated channels. Since then, Dell Computer Corp. has become known as the world's leading direct-selling computer-system company. [37]

Dell's markets softened in the early 1990s, and the industry went back to a technology focus: marketing the fastest chip, the fastest bus architecture, and the best bundle of software. By 1995, though, Dell's business took off again (based on the same model).

The degree to which competitors in the PC industry succeed or fail is ever more dependent on supply-chain efficiency. Many participants still focus mainly on products and markets and neglect aggressive management of the supply chain itself as a strategic source of value. In addition to answering questions about component cost and availability, companies must consider time-to-market and distribution methods and the supply chain's ability to function efficiently and respond to changing customer preferences throughout the product's life cycle. According to Terrence Austin, an associate partner at Anderson Consulting, "The competitive battlefield for the next five to ten years in the PC industry is going to be much more focused on service-delivery, product placement, and inventory efficiency (i.e., supply-chain efficiency), than it is on product performance." [37]

The Changing Distribution Model for the PC Industry

The PC distribution model is different today. Direct-sellers are winning market share from "indirect" sellers, those that sell through retail stores and business-oriented "resellers," for three reasons: price, value and service.

Direct PC sales this year are expected to account for at least one-third of the 37.1 million PCs sold in the U.S. The direct market is split literally among hundreds of players. Still, three companies --- Dell, Gateway and Micron --- control two-thirds of the direct-sales market.

Dell computer's marketing strategy is to sell directly to customers, instead of going through resellers and distributors. This strategy contrasts with the approach of the PC industry's big three --- Compaq, HP and IBM --- whose distribution deals make them vulnerable to dealer blackmail. Compaq, HP and IBM are seeking to beat Dell by refining their three-step distribution model, from manufacturer to distributor to reseller to customer. Refer to Figure 1.1.

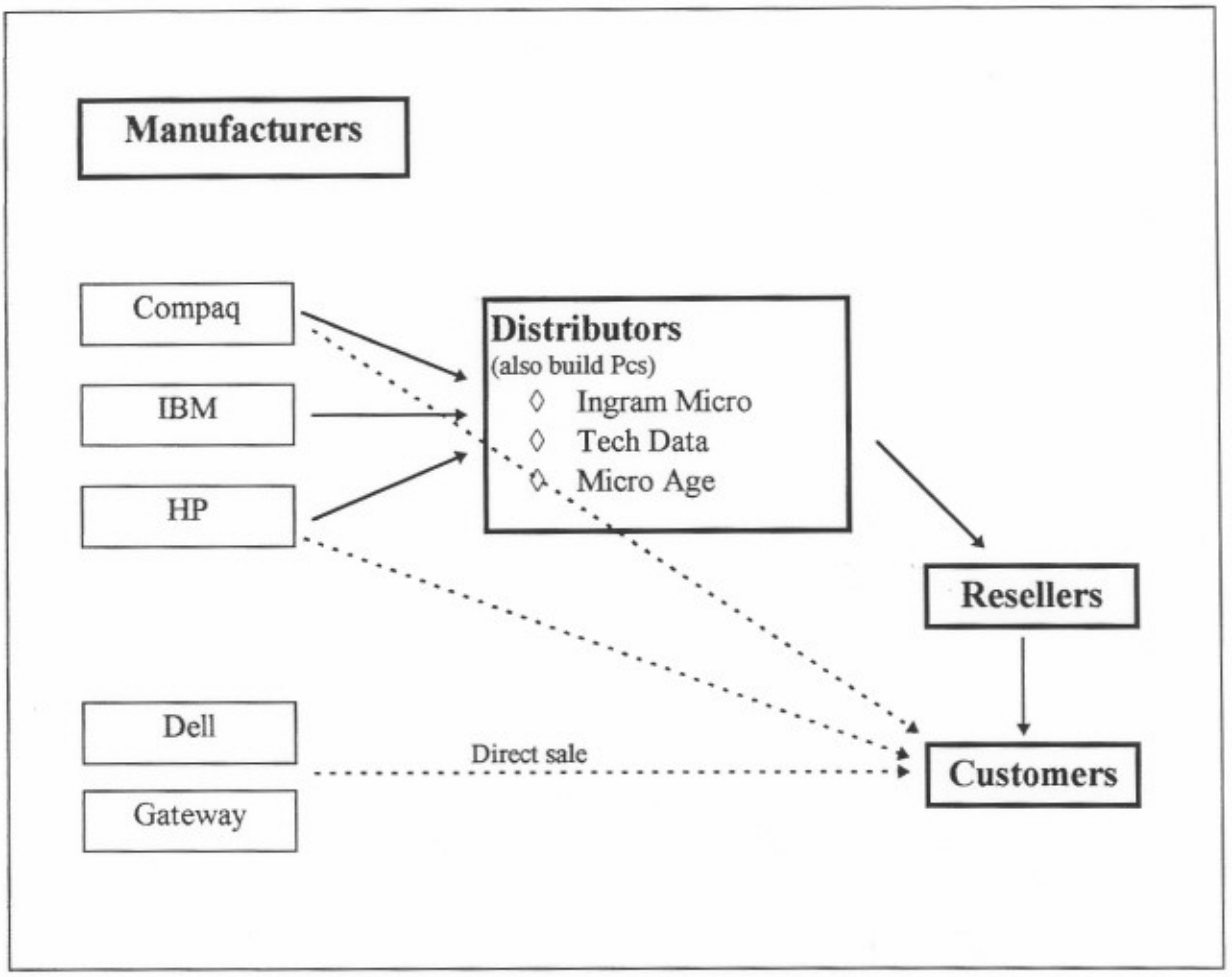


Figure 1.1 The new distribution model.

Individual Strategies in the PC Industry

Dell

Dell is concentrating on tech-savvy consumers, believing that owners of home computers are knowledgeable about the computer's components and performance. Lately, Dell has been growing about six times as fast as the consumer market. In the past, Dell was not interested in the computer market because (1) its margins were too low and (2) novice computer users required large amounts of margin-burning technical support after the sale. [27]

Today Dell believes that consumers are ready to order big-ticket items directly through the Internet. Surveys indicate that the Internet is a key reason for about three-fourths of all PC sales. Dell signs agreements with companies that provide cable access as well as high-speed

access over telephone lines using ADSL technology to help its business grow. [14] Dell also increased to three years of service on its new home computers. [6]

Gateway

Gateway historically concentrated on the home market, and Gateway is now trying to steal Dell's business customers. [27] In October 1998, mail-order giant Gateway introduced a part-exchange scheme. Gateway will deduct the second-hand value of any computer it has previously sold from the price of a new PC as long as a customer buys through Gateway's finance scheme or purchases software. [13] This program, Your:)Ware™, enables customers to trade in their two year old PC for a new one. In this market of ever changing computer technology, this program gives a large sense of security to the consumer. [38]

Compaq

In order to increase revenue and market share, Compaq is going to a Dell-like build-to-order model. [27]

IBM

IBM has drawn up a strategic blueprint to transform its enterprise business into an electronic commerce powerhouse that spans millions of corporate servers and desktops. The blueprint, which is actually an internal document for IBM's Network Computing Software Division, calls for IBM to "bet the bank on Java, pump up Netscape's business to head off Microsoft's Internet Explorer, and aggressively push OS/2 customers onto other platforms." [20]

PC Trends for the Future

To get an indication of what are some of tomorrow's trends in consumer computer products, one can take a look at what innovations are being introduced at this year's Comdex show. [21] The explosion of the Web and the prospect of myriad "information appliances" connected to it has led pundits to proclaim the beginning of the "post-PC era." Judging by history, the transition could still take awhile and will not necessarily spell doom for the PC. The mainframe computer business, after all, is still growing 20 years after the PC was introduced. This year about $217 billion of PC hardware will be shipped world-wide, and PC unit shipments are growing at a robust rate of about 13% to 14%. Market researchers expect that unit growth to continue at double-digit rates for some years to come, partly due to new sub-$500 PCs being introduced at Comdex in November 1998. There is no doubt that the PC business is maturing, and hefty profits are hard to come by except for a handful of players such as Microsoft, Intel, and Dell. Hundreds of companies, ranging from tiny start-ups to telecommunications giants, are turning to the appliance market in hopes of finding new growth and profits and escaping the death grip of Microsoft and Intel on most PC-related wealth.

"The shift in attitude is that the appliance is no longer a toy," says William R. Sell, Comdex's general manager. According to International Data Corporation (IDC) of Framingham, Massachusetts, such devices could begin to match the PC's unit sales a few years from now. A recent IDC study shows that 3.4 million appliances --- mainly hand-held computers, smart phones, video-game players linked to the Internet and digital set-top boxes --- will be shipped in the U.S. this year, compared with 36 million PCs. "Their convenience is the key attribute," says Frank Gens, Senior Vice-President of Internet Research at IDC. "Right now in any home, the PC is in a room in the house that may or may not be convenient, while appliances will be available depending on how and where you want to use them, from email to shopping to getting news." [21]

Gens adds that as more of these devices come into use, more applications will be written for them and their utility will increase. He also believes that many of them could eventually cost nothing as companies give them away to charge for the more lucrative services they access. 3Com Corp's Palm Pilot hand-held computer, with one million units sold in 1997, exemplifies the appeal of the newfangled appliances. It has given birth to a large number of software companies whose programs take advantage of it by accessing electronic mail, web sites or phone numbers for people on the run. Dubinsky, who headed 3Com's Palm Computing Inc., predicts that appliances that link to the Internet will be the next frontier and could eventually surpass the sales of personal computers. "They add a whole new level of basic functionality that is not redundant to the PC . . . and they are easy to use, adding a clarity of purpose," she says. [21]

Operational Strategy

The focus within Operational Strategy is on the application of and the benefits experienced through reuse. Specifically, information is provided as to why companies should apply this concept and how it can be implemented. Case studies give success stories for the reader.

What is reuse?

Reuse is a ***strategic*** practice of systematically planning the creation and application of reusable components. Reuse and modular design should be an integral part of any engineering discipline. As an example, mechanical engineers do not design an engine from scratch for each car model that is introduced on an assembly line. The architecture and design of an item is reused to develop and manage a product line in any engineering discipline and most prevalently in the computer industry. Unfortunately, many companies do not follow this practice.

The concept of reuse in design has both technical and organizational issues that require careful management techniques for successful implementation. The process involves several key strategies: transition of the organization's processes, methods and tools based on the reuse principles; minimization of the risks associated with the insertion of reuse requirements into the activities and projects in the company; and employing systematic reuse rather than opportunistic reuse. [9]

The Benefits of Reuse

In an era of increasing competition on a global scale, management and technical skills are challenged to improve the reliability and quality of systems. In addition, systems have ever increasing complexity and sheer size. A successful and proactive response to this situation is the implementation and integration of software/hardware reuse principles to develop and maintain systems over their entire life-cycle. The derived benefits are interoperability, rapid prototyping, shorter overall life-cycles, less initial risk, higher-quality products, and lowered long-term costs. [30]

What to Reuse

In the case of software development, only a small fraction of the time dedicated to a project is consumed by coding. Yet almost all reuse efforts have been geared towards code reuse. A broader approach must be emphasized by also advocating the reuse of items such as designs, specifications, user documentation, and concepts of operation.

Applying the Concept of Reuse

The foundation of a reuse effort is disciplined and well-defined methodology that addresses a complete life-cycle; develops mechanisms to manage, trace and grade through the life-cycle; and sets milestones in the life-cycle [9]. This is essentially emphasizing that the reuse effort must be managed well with the fundamental principles of project management in mind.

A key activity associated with reuse program creation is domain engineering which is a systematic way of identifying potentially reusable assets and defining an appropriate architecture to enable their reuse. The creation of reusable components is typically more expensive than "ordinary" development, and hence development for reuse needs to be justified for long-term economic and business reasons. Reusable components must anticipate future needs. The result of domain engineering is a grouping of common and variant features of the company's assets and providing for "certified" reusable system components.

The management of the reuse effort should be focused on a "product line" concept rather than a single software system. The product line approach allows for the development of many products from one architecture. Personal computers are an example of significant component reuse within a product line.

Reuse methodology must be an internal company philosophy. However, a company does not need to solely rely on internal mechanisms to maintain reuse, but can utilize outside resources for areas where they do not have knowledge or capabilities. There are several independent companies that provide such services. An example is Aspec Technology (California) which is a leading provider of semiconductor intellectual property libraries and design services. This company provides verifiable libraries and design implementation methodologies to enable companies to minimize re-design time. The customer's initial design development costs and investments could be amortized over multiple product generations.

Jacobson, Griss and Johnson [22] observed the application of reuse in several organizations including Hewlett-Packard. The benefits that the companies obtained from implementing reuse were improved time-to-market and quality, lower overall development costs and increased levels of reuse with maturity of the program. The authors noticed an incremental adoption of reuse in these companies that indicated trends as shown in Figure 2.1. These companies experienced improved benefits as they progressed from one stage to the next.

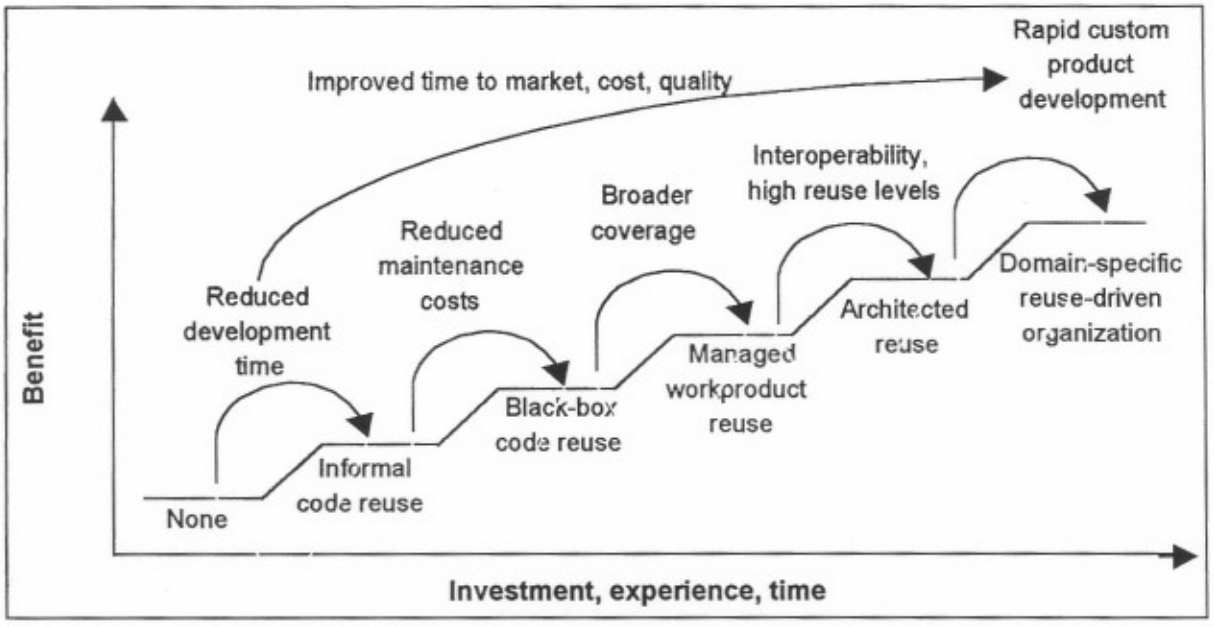


Figure 2.1 The incremental adoption of reuse.
This figure is based on a synthesis of experience at several organizations within Hewlett-Packard and elsewhere.

Moving from no reuse to "informal code reuse" is generally what individual developers naturally stumble onto by modifying sections of code that they may have utilized in the past. These developers are at times informally familiar with code written by others. This requires programmers to have a high level of trust in each other. In order for the organizations to move towards "black-box code reuse," the reusable modules were reengineered, tested, and documented. The modules in this phase were utilized without modification, but still not on a company-wide basis. After the number of reuse projects increased, the companies moved towards "managed workproducts" dealing with managed standard versions, maintenance, testing, adaptation, training and education. In order for reuse to more broadly cover the organization, companies moved to "architected reuse" which accounted for correlation and compatibility of components with a common architecture. The final step that Jacobson et al. [22] noticed was toward a structured and driven reuse program in a company that involved developing domains as focused parts of the complete product architecture. This enabled rapid custom development in a cohesive manner within the organization to meet diverse customer requirements.

The development approach should be domain-specific and architecture-centric in order to provide for a reuse system that can be managed, is reliable and stable, and can be enhanced. This is to say that the modules must be designed and categorized in specific domains, but must still maintain the company wide architectures. Reuse can be accomplished by utilizing a

variety of technologies such as object-oriented languages, standardized computer-aided design tools, repositories, and libraries.

There are several up-front investments that must be made to develop a reuse system. These involve (1) creating a separate group of domain engineers to analyze the boundaries of systems and develop architectures; (2) developing and maintaining asset management tools such as repositories for architectures, code, designs, and documentation; and (3) providing tools to aid design and to speed prototyping, modifications and maintenance.

One of the most common failures in the process of transitioning to reuse is insufficient investment and top-level commitment allocated to the effort. This, as with any project, will decrease the likelihood of success. Proactive leadership must be provided at the beginning of development projects to encourage reuse. A separate group must be created within the organization whose sole job is to create, maintain, and provide accessibility to reusable products.

Productivity metrics in an organization generally emphasize products developed and do not give weight to functionality. A rule-of-thumb for development of reuse products is that, in the short term, they will take twice the effort of a one-shot product [9]. Hence, the tendency is toward the lower cost process. In order for reuse to succeed, the emphasis should be placed on enterprise productivity across a product line of systems, rather than at individual productivity levels. Contract fees should be based on providing incentives to contractors to populate architectures and repositories. Companies must incorporate incentives in their contracting for the extra effort required.

Reuse must add value to the project and not contaminate it. Reuse should complement and aid in the achievement of business and engineering goals.

Case Studies

Ericsson

In 1967, this Swedish based company wanted to supply telecommunication switching systems worldwide. They faced the challenge of having to adapt their systems to several countries. Several key architecture, process, and organizational principles can be outlined [22].

Initial investment and long-term organizational commitment. In the beginning, some people in the company made a strategic decision to design and implement an architecture that provided for substantial reuse. The decision to move towards this strategy was advocated by

a technical product manager who served as a technical champion for this concept. It took three years of development for the first useable product that had limited commercial success. This initial product was then upgraded to a much more flexible and maintainable product base system that enjoyed significant commercial success. It took almost ten years from the original decision before most of the people at Ericsson recognized that the key to the success was the architecture and development processes based on reuse.

Well-designed Architecture. Ericsson developed an architecture aimed at flexibility and ease of adaptation for reuse and customization. The system architecture was articulated and documented very carefully to provide an understandable framework. This not only served to keep everyone within Ericsson on the same page but it also enabled the salespeople to explain the functioning of the system to prospective customers. The system was flexible and adaptable to meet the needs of most customers.

Modular to Enable Various Configurations. Ericsson implemented its system by melding carefully designed small subsystems that they called service packages. These packages had well-defined interfaces and tested compatibility with other packages. The company was able to make about 90% of its service packages completely reusable and with attractive features to minimize the amount of customer specific components that would be requested.

Ericsson's product is believed to be the largest product ever built using object-oriented technology [22].

Motorola

In the early 1990s, Motorola took steps to initiate a reuse strategy to increase productivity and improve product quality. Motorola's first step in this initiative was to form a "reuse task force" which comprised leading technicians from all sectors in the company and a "reuse working group" [24]. The task force's goal was to take abstract reuse goals and develop and recommend detailed process guidelines on reuse practices. They recommended a two-fold approach of collecting information on the past requirements, designs and code, and placing these in viable reuse databases, and also focusing on future product designs. They conveyed the importance of the reuse concept to top management and also recommended a more organized and extensive reuse effort with orientation and training. The volunteer programs that evolved from this had limited success, but it was not enough to convince middle-management of the up-front costs.

It was necessary for Motorola's CEO to get involved to make upper-management recognize the significance of this concept. Motorola included the reuse concept in its business strategic plan for the future. A cash-incentive award program was implemented to reward developers

each time their component was retrieved from the database for reuse. The program also rewarded a user for reusing from the database. As a result of this implementation, significant gains in productivity and quality were measured within the company. With this success, Motorola is developing an environment that is reuse-friendly and further developing formal reuse toolkits that standardize, if not automate, the reusable software development cycle.

Synopsys

The ability to reuse Intellectual Property (IP) in chip design has become mandatory for today's feature-rich, complex applications. The need for versatile IP reuse results from rapidly growing complexity of system-on-a-chip designs. By year 2000, chips consisting of 12 million gates will be commonplace. Based on today's metric of 100 gates per day to design a new chip, it would take 400 person-years to design each of those chips [36]. Reuse promises to bring design cycles into a more realistic time frame.

Synopsys has developed technology libraries that deliver silicon-proven IP that is portable across many semiconductor fabrication processes. With this, designers can quickly and efficiently integrate IP into new designs and processes.

Every company's set of reuse challenges is unique. Synopsys has taken the approach of studying the company's product portfolio, organization and current design processes and then creating customized solutions that balance near-term and long-term objectives. Synopsys offers a full complement of services and technology to solve design reuse problems at different levels of scope and complexity. Some of the services are:

- Classroom training on reuse methodology and the fundamentals of how to create reusable designs.
- Customers identify the IP and then Synopsys converts them to reusable form.
- Helping customers create a design reuse infrastructure within their companies. This is for industry players ready to make design reuse the fundamental paradigm driving their product development strategy.

Engineering/Product Life-Cycle

In response to fierce competition within the computer industry, companies are working to shorten product life-cycles and decrease product cost.

The Problem

As previously mentioned, the product development environment of the computer industry is undergoing an increasingly volatile period of change where flexibility will dictate the winners and the losers. Unfortunately, most of the management practices standard in the industry today are characterized more by their rigidness and centralization rather than their flexibility. From an engineering process approach, much of this is exacerbated by the rigid, serial nature of the process and life-cycle technologies most often used in today's computer industry.

Although things are beginning to change, the current standard for engineering process is the traditional waterfall life-cycle. In the waterfall model (Figure 3.1), a project progresses through an orderly sequence of steps from the initial concept through product testing and manufacturing [26].

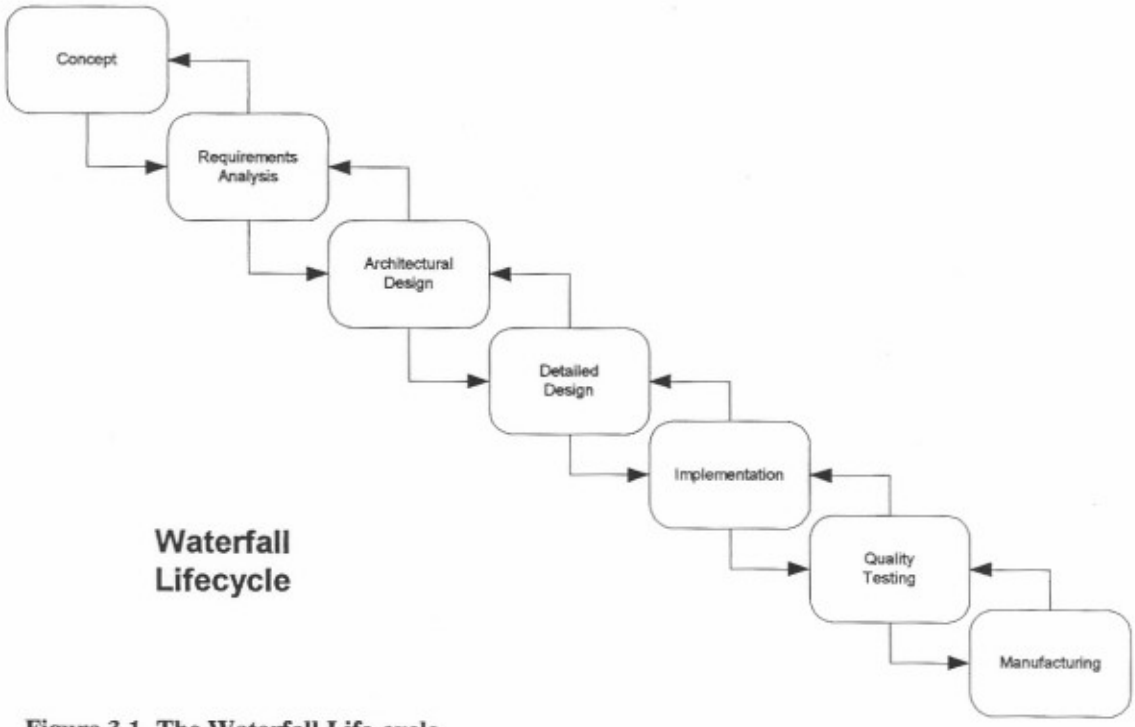


Figure 3.1 The Waterfall Life-cycle.

This sequence of steps occurs almost exclusively as a non-continuous serial process where each step is fully completed before the following step is allowed to begin. While perfectly valid for environments with stable product definitions and well-understood methodologies, the waterfall life-cycle has several serious shortcomings in today's dynamic computer industry. It forces a serial approach to problems that are not completely sequentially linked. Implementation prototypes could easily begin before formal design is complete and similarly testing and manufacturing planning could easily begin before implementation is complete. This serial process unnecessarily lengthens the development cycle and also limits communication flow among the various groups involved. This communication throttling can easily be seen in the lack of contact with the customer anywhere but at the beginning and end of the development cycle. This lack of contact with the customer results in a lack of confidence on the customer's part and an increased risk of producing something other than what the customer wants. Such inflexibility is a recipe for disaster in today's fluid environment. Obviously new more flexible approaches are required.

Solutions

Analysis of available literature and case studies yields two emerging examples of this trend toward flexibility in the engineering process arena of the computer industry: concurrent engineering and iterative & incremental product life cycles. Concurrent engineering can be established at several levels across projects and activities and is often integrated with an iterative & incremental approach to multiply the benefits of each solution [31, 18, 16].

Concurrent Engineering

In 1987, the US Defense Advanced Research Projects Agency (DARPA) set up a group of academics, industrialists and government specialists to examine the implications of simultaneous engineering for defense sourcing. They called the concept of parallel processes 'concurrent engineering.' They also provided another definition for the Manufacturing community: "Concurrent Engineering is a systematic approach to the integrated concurrent design of products and their related processes including manufacturing and support. This approach is intended to cause the developers from the outset to consider all elements of the product life cycle from conception through disposal including quality, cost, schedule and other user requirements." [2]. The main objective of concurrent engineering is to shorten the time-to-market while maintaining the highest quality, lowest cost, and customer satisfaction. Reduction of lead-time is achieved by parallel development and production activities [10]. Concurrent engineering methodologies have been widely adopted and with success especially in hardware development through simultaneous performance of activities and processing of information [3, 4]. Hewlett Packard, Chipcom, Sun Microsystems, and Digital are among many who have reported significant product life-cycle improvements after concurrent engineering was implemented. [34] Companies from other industries such as Boeing and Texas Instruments have also benefited from Concurrent Engineering. [35]

While concurrent engineering offers obvious benefits to process efficiency, it also requires some modification of organization structure. For proper application it demands that the traditional functionally based organization structures be done away with and the new concept of multifunctional teams be introduced. Multifunctional teams are development groups made up of representatives of all the functional roles played in the development effort. They approach each step of the development process as a united front thereby maximizing information flow throughout the development effort and enabling work to begin in all areas as soon as it is possible. Durand [8] studied the dependencies between levels of concurrency and how the project's groups function in the organization. One of his observations was that the teams had to be multifunctional or the organization had to be restructured to accommodate the multifunctionality to have a concurrent environment. And the companies he studied in the computer industry were displaying significant concurrency. Figure 3.2 shows an example of concurrent engineering integrated with an incremental development life-cycle.

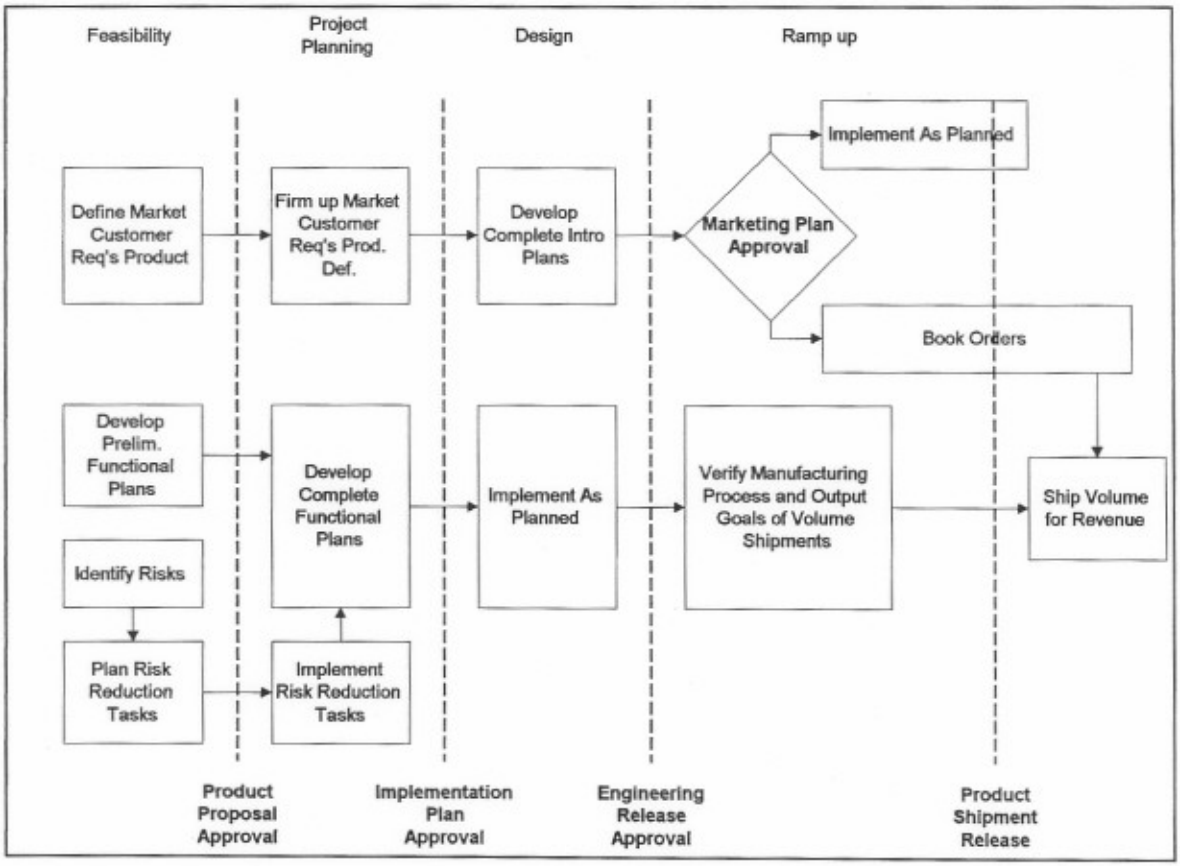
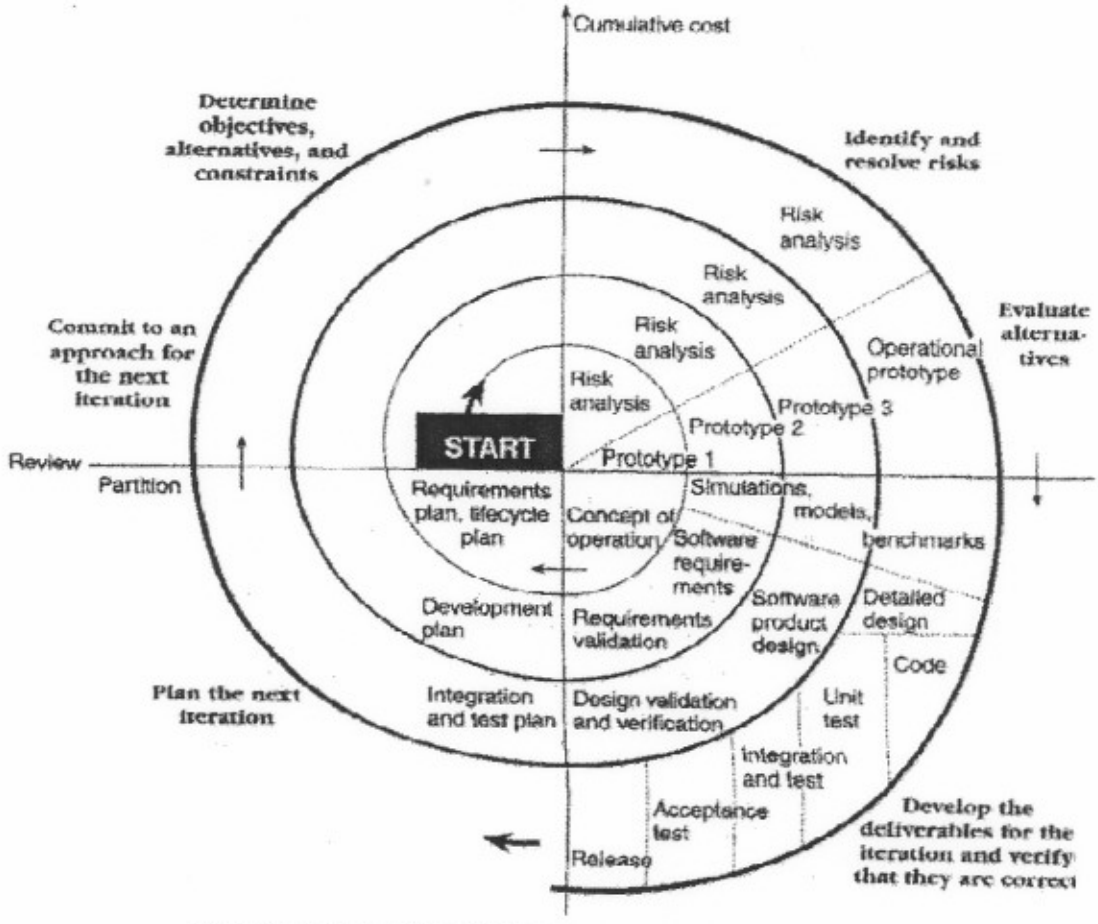


Figure 3.2 Concurrent engineering integrated with an incremental development life-cycle.

Iterative and Incremental Development Life-cycle

The concept of an iterative and incremental development life-cycle is to avoid the pitfalls of a dynamically changing environment by breaking the development task up into many smaller chunks. Where the traditional waterfall life-cycle requires that the problem needs to be fully understood and defined up front, the iterative and incremental life-cycle states openly that this is impossible in a dynamic environment and instead embraces the changing nature of the problem. The iterative and incremental approach solves the problem by tackling individual chunks of the problem one at a time starting with the most critical piece first. Each increment is executed using a modified waterfall approach that often leverages concurrent engineering, and is completed by a reassessment of the problem as a whole. This reassessment includes feedback to the customer, which is now available very early in the project, as well as inclusion of any new or modified project requirements and planning of the remaining project increments.



Source: Adapted from "A Spiral Model of Software Development and Enhancement" (Boehm 1988).

Figure 3.3 A Spiral Model of Software Development and Enhancement.

In this manner, the entire process is driven by risk reduction such that the farther along the project progresses the lower the risk becomes. In opposition, the traditional waterfall approach does not inherently consider risk and is purely document driven such that as the farther the project progresses the higher the risk that the end product will not actually meet the customer requirements. This incremental approach not only reduces risk but also by uncovering problems early in the life-cycle it also tends to increase quality and reduce overall development cycle times. While the iterative and incremental life-cycle was originally developed in the software world, its risk driven incremental approach is just as applicable to all other segments of the computer industry. From an engineering process viewpoint, an iterative and incremental life-cycle implemented using concurrent engineering practices and multifunctional teams is the current state of the art in flexible management practices. [26]

Concurrent Engineering

This section contains examples of companies that have successfully applied the practice of concurrent engineering.

Motorola

Motorola was one of the pioneers of concurrent engineering. The applied practice of concurrent engineering is now a foundation technology across almost all company divisions and business units.

NEC

NEC Japan has attained successful results through a series of strategic commitment to capabilities distributed throughout its network of suppliers. They have beaten Fujitsu and IBM Japan in the Japanese Market. [28] Their computers were not first to market, however they met the needs of the greatest number of consumers. One of the reasons of their success can be that they understood what the customers required, meaning that they were able to connect engineering and marketing functions, so that they would work concurrently through the product development effort.

IBM

IBM UK which is developing and manufacturing disk drives, brought development, manufacturing, marketing, and after sales service together to work as a team in order to shorten its product development cycle. They succeeded through concurrent engineering.

Iterative and Incremental Development

This section contains examples of companies that have successfully applied the practice of iterative and incremental product development.

Netscape

The company effectively executed a textbook iterative and incremental approach in the development of their Netscape browser. Development began with an initial shell of core functionality which they quickly put up on the company Intranet for employees to play with and give "customer" feedback. This feedback was collected, analyzed and implemented through many functional iterations. Once the software had reached a stability level that it was ready for real customer feedback, Netscape put its browser out on the Internet as a free beta product. The product went through a similar series of iterations here and was eventually released to great success. Due to its unmitigated success in creating a product that the customer truly wanted in an amazingly short timeframe, this development process is now the standard process at Netscape and most other Internet software companies. [16]

Microsoft

Microsoft after suffering initial defeat at the hands of Netscape quickly realized the potential of Netscape's new process and implemented a very similar strategy for their Internet Explorer browser. The only real difference was that Microsoft with a much bigger name to protect was not nearly so eager as Netscape to place beta quality software out on the Internet. Instead they relied almost exclusively on their internal company Intranet. While depriving them of the broad voice of the real customers, their effort was nonetheless successful mainly due to Microsoft's large employee base of 18,000, most of whom are accomplished websurfers and understand the issues at hand. [16]

Yahoo

Yahoo was another company who realized early on the benefits of the iterative and incremental approach especially in the dynamic world of Internet software where not only your application requirements but the very platform they run on changes almost daily. They followed a similar pattern as Netscape and Microsoft in iterating first internally and then externally. Yahoo released the new versions of their new services on-line for internal use only. And later on they ask about 30,000 users who have volunteered to be beta testers, to try the new service. They also used flexible engineering resources to pull from other areas and deploy into needed areas when necessary. [16]

Summary

In order for a company to be competitive in the computer industry today, its management and engineering practices must be flexible enough to respond to customer demands, yet disciplined enough to decrease time to market, lower product costs, and still ensure high quality. To manage this environment, several companies are changing the way they market and develop their products.

Within marketing strategy, there is a move away from engineering-driven product development in which case a company builds a product based on what the company believes the market will want. Instead, there is a move toward customer-driven product development in which case the company seeks to understand what its customers want and need; then, based on the information acquired, the company builds a product to meet those customer needs.

Within operational strategy, there is a move away from independent, custom development in which each project and product is built from the ground up. Rather, there is a move toward modular, reusable development in which projects and products are planned and designed in such a way that effort and resources expended on one project or product can be reused on other projects and products.

There is also a move away from rigid engineering processes, such as the waterfall approach to software development, and a move toward more flexible engineering processes, such as concurrent engineering and iterative and incremental development.

The trends presented in this paper represent only a portion of what is happening in the industry. Still, they give the reader a flavor of the computer industry today and the computer industry of tomorrow.

References

[1] "Apple's IMac Top-Selling Desktop PC in Computer Superstores," *Dow Jones News Service*, November 12, 1998.

[2] C.J. Backhouse and N.J. Brooks, *Concurrent Engineering*, Wiley, 1996.

[3] J.D. Blackburn, G. Hoedemaker, and L.N. Van Wassenhove, "Concurrent Software Engineering: Prospects and Pitfalls," *IEEE Transactions on Engineering Management,* vol 43, no 2, pp. 179-188, 1996.

[4] D.E. Carter and B.S. Baker, *Concurrent Engineering: The Product Development Environment for the 1990s*, Addison-Wesley, 1992.

[5] "Customer-Driven Marketing" (October 8, 1998): Online. Available WWW: http://www.collegegrad.com/book/1-4.html.

[6] "Dell's Dimension," *Computer Weekly*, October 8, 1998.

[7] L. Demester and C.S. Tang, "Reducing Cycle Time at an IBM Wafer Fabrication Facility," *INTERFACES,* vol 26, no 2, pp. 34-49, March-April 1996.

[8] T. Durand, "Concurrent Engineering and Interfunctional Project Groups," *International Journal of Technology Management*, Vol 10, No 1, 1995.

[9] "DoD Software Reuse Initiative" Available WWW: http://dii-sw.ncr.disa.mil/reusic, 1996.

[10] W. Eversheim, A. Roggartz, H.J. Zimmermann, and T. Derichs, "Information Management for Concurrent Engineering," *European Journal of Operational Research 100*, pp. 253-265, 1997.

[11] T.D. Floyd, S. Levy, and A.B. Wolfman, *Winning the New Product Development Battle,* IEEE Press, 1997.

[12] D. Gatenby, "Design for 'X': Key to Efficient, Profitable Product Realization," *Quality Year Book*, Mc Graw Hill, 1997.

[13] "Gateway Launches Used-Computer Trade-In Scheme," *Knight-Ridder/Tribune Business News*, October 6, 1998.

[14] J. Hawkins-Mahoney, "Dell Computer Chairman Says Faster Internet Access Will Benefit Consumers," *Knight-Ridder/Tribune Business News*, October 12, 1998.

[15] R. Hayes, G. Pisano, and D. Upton, *Strategic Operations*, Harvard Business School, 1996.

[16] M. Iansiti and A. MacCormack, "Living on Internet Time: Product Development at Netscape, Yahoo, NetDynamics, and Microsoft," *Harvard Business School Case Study*, 9-697-052, April 1997.

[17] M. Iansiti and A. MacCormack, "Developing Products On Internet Time," *Harvard Business Review*, pp. 108-117, Sep-Oct 1997.

[18] M. Iansiti, *Technology Integration*, Harvard Business School Press, 1998.

[19] M. Iansiti and J. West, "Technology Integration: Turning Great Research into Great Products," *Harvard Business Review*, pp. 69-79, May-June 1997.

[20] "IBM renews battle for desktop," *PC Week*, October 5, 1998.

[21] "Information Appliances Invade PC's Turf," *The Asian Wall Street Journal*, November 13, 1998.

[22] I. Jacobson, M. Griss, and P. Johnsson, "Software Reuse: Architecture, process and organization for business success," Addison-Wesley, 1997.

[23] B. Johnson, "Direct sellers taking business from traditional PC suppliers: Dell powers past Compaq in mid-1998 computer sales," *Advertising Age*, September 28, 1998.

[24] R. Joos, "Software Reuse at Motorola," *IEEE Software*, pp. 42-47, September 1994.

[25] J. Magretta, "The power of Virtual Integration: An Interview with Dell Computer's Michael Dell," *Harvard Business Review*, pp. 73-84, March-April 1998.

[26] S. McConnell, *Rapid Development: Taming Wild Software Schedules*, Microsoft Press, 1996.

[27] D. McGraw, "The tech-know generation: Dell takes aim at increasingly sophisticated home PC market," *U.S. News & World Report*, October 12, 1998.

[28] D.T. Methe, R. Toyama, and J. Miyabe, "Product Development Strategy and Organizational Learning: A Tale of Two PC Makers," *The Journal Of Product Innovation Management*, vol 14, no 5, pp. 323-336, 1997.

[29] K. Pawar and J. Riedel, "Achieving Integration Through Managing Concurrent Engineering," *International Journal of Production Economics*, pp. 329-345, 1994.

[30] J.S. Poulin, J.M. Caruso, and D.R. Hamdcock, "The business case for software reuse," *IBM Systems Journal*, vol. 32, no 4, pp. 548, 1993.

[31] B. Prasad, *Concurrent Engineering Fundamentals Volume II,* Prentice Hall, 1997.

[32] J. Ritter, "Customers Drive Future of Electronic Commerce," (October 8, 1998): Online. Available WWW: http://darkwing.uoregon.com/~pacc/archives/second_issue/ritter.html.

[33] R. Schmidt, "The implementation of simultaneous engineering in the stage of product concept development: A process orientated improvement of quality function deployment," *European Journal Of Operational Research* 100, pp. 293-314, 1997.

[34] S.G. Shina, *Successful Implementation of Concurrent Engineering Products and Processes*, Van Nostrand Reinhold, 1994.

[35] M.L. Swink, J.C. Sandvig, and V.A. Mabert, "Customizing Concurrent Engineering Processes: Five Case Studies," *The Journal Of Product Innovation Management*, vol 13, no 3, pp. 224-244, 1996.

[36] "Synopsys Reuse Manual" (November 16, 1998): Online. Available WWW: http://www.synopsys.com.

[37] J. Teresko, "The New PC Game: Will design-for-responsiveness win the supply-chain battle?," *Industry Week*, pp. 50-53, May 18, 1998.

[38] "Your:)Ware" (November 19, 1998): Online. Available WWW: http://www.gateway2000.com/home/yourware/.